



CARMML

A Pretty Sweet RML Engine

Victor Louwerse & Jan Voskuil

PLDN

6 December 2017, Amersfoort







- Open Source created by Taxonic, with Kadaster
- RML standard for mapping & transformation
- From relational to RDF
- From any format to RDF
- **Now** available from Github!
- <https://github.com/carmil>



carmil

AGENDA

Bedrijfsperspectief

Ontwerp

Gebruik

Roadmap

Probleemstelling

- ❖ Hoe krijg ik **legacy-data** in mijn Linked Data-omgeving?
- ❖ Scriptjes, pipelines
- ❖ Schaalbaarheid, beheersbaarheid
- ❖ Behoefte: **standaardaanpak**

Probleemstelling

- ❖ Hoe krijg ik **legacy-data** in mijn Linked Data-omgeving?
- ❖ Scriptjes, pipelines
- ❖ Schaalbaarheid, beheersbaarheid
- ❖ Behoefte: **standaardaanpak**

- ❖ Daarom: RML als specificatie
- ❖ Voorbeeld: R2RDF (W3C)
- ❖ Mapping vastleggen in een RDF-graph
- ❖ Gebruik makend van een standaard vocabulaire
- ❖ Uitgebreid naar (bijna) alle mogelijke bronformats

Voordelen RML

- ❖ Herhaalbaar, beheersbaar
- ❖ Declaratief in RDF
 - ❖ Inzichtelijk
 - ❖ Eenvoudig te leren
 - ❖ Voor analisten en ontwikkelaars
- ❖ Schaalbaarheid
- ❖ Gestandaardiseerd, dus herbruikbaar

Waarom een nieuwe implementatie?

- > Kadaster stelt **hoge** eisen
- > Bestaande implementatie voldoet niet
- > Nieuwe implementatie nodig
- > TAXONIC en Kadaster doen dit samen
- > Resultaat is Open Source

Open Source, wat betekent dat?

- > **Managed** Open Source
- > Support beschikbaar
- > Softwarekwaliteit wordt **bewaakt**
- > Ondersteuning voor community
 - documentatie
 - cursussen
 - meetups

What is RML?

```
id, stop, latitude, longitude  
6523, 25, 50.901389, 4.484444
```



```
<http://airport.example.com/6523> a transit:Stop ;  
    transit:route "25"^^xsd:int ;  
    wgs84_pos:lat "50.901389" ;  
    wgs84_pos:long "4.484444" .
```

What is RML?

```
rr:subjectMap [  
  rr:template  
  "http://airport.example.com/{id}";  
  rr:class transit:Stop  
];  
  
rr:predicateObjectMap [  
  rr:predicate wgs84_pos:lat;  
  rr:objectMap [  
    rml:reference "latitude"  
  ]  
];
```

CARML

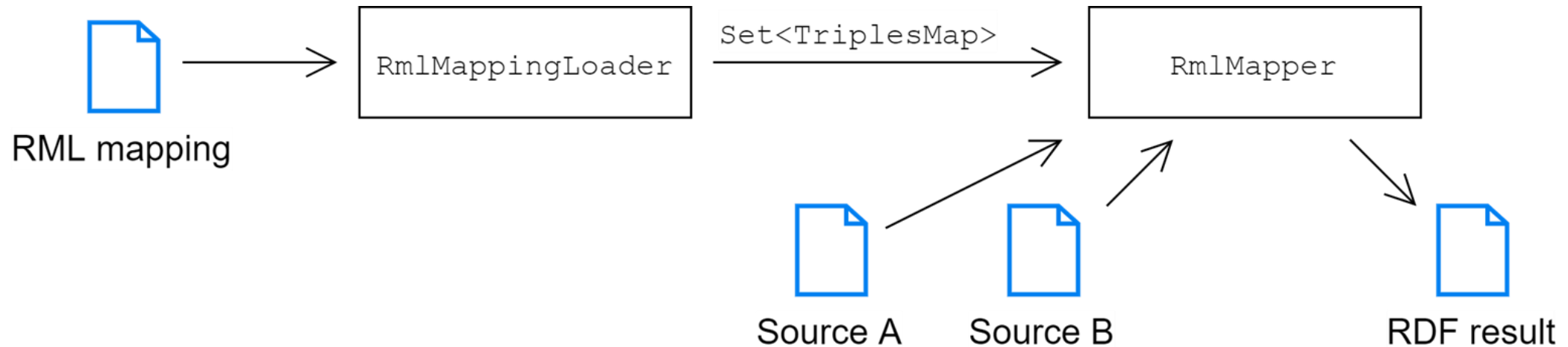
RML engine/library

Features

- Engine supports full RML implementation
- Actually works
- Fast for most cases
- Function support through FnML

CARML design

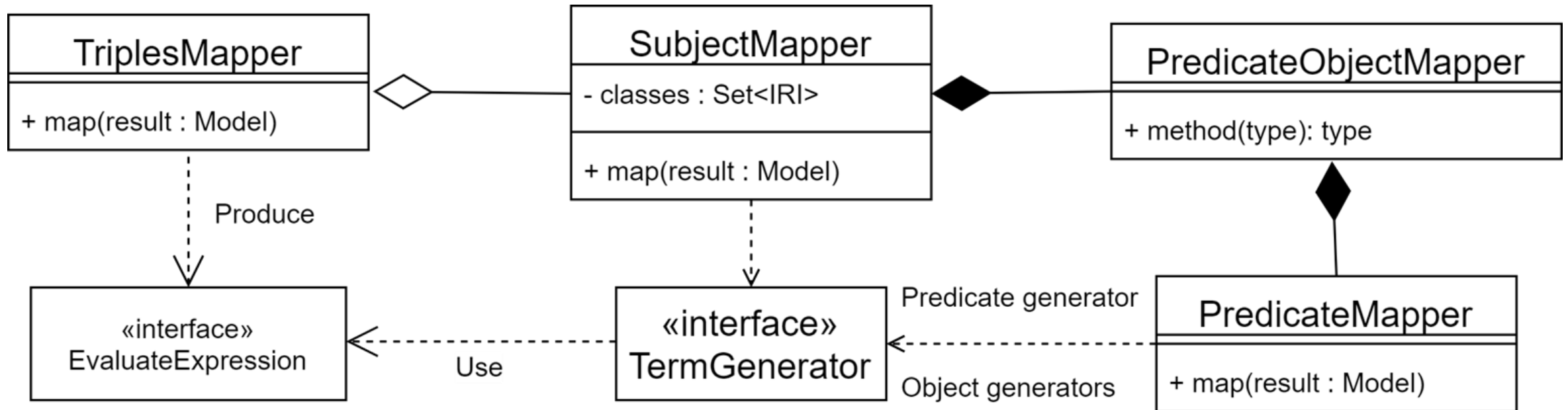
- Modules
 - RDF-to-object loader
 - RML object model
 - RML engine



CARML design

Classes involved in actual mapping by engine.

Suitable for re-use across repeated mappings.



CARML design

Define functions with annotations

```
@FnoFunction(Ex.toBoolFunction)
public boolean toBoolFunction(
    @FnoParam(Ex.yesNoString) String yesNoString
) {
    return yesNoString.toLowerCase().equals("yes");
}
```

```
mapper.addFunctions(new MyFunctions());
```

Using CARML

```
<dependency>  
  <groupId>com.taxonic.carm1</groupId>  
  <artifactId>carm1</artifactId>  
  <version>0.1.0</version>  
</dependency>
```

```
RmlMapper mapper =  
    RmlMapper  
        .newBuilder()  
        .fileResolver(Paths.get("my/data/sources"))  
        .build();  
Model result = mapper.map(mapping);
```


Next steps

- Optimizations for mappings with many “joins”
- Graphical editor for RML mappings

